Seminar 2: Cryptography

Jonathan Ashmore, Team Availability

Task

- Read the Cryptography with Python blog at tutorialspoint.com
- Create a python program in VSC that can take a text file and output an encrypted version as a file in your folder on your VSC system.
- Demonstrate your program operation in this week’s seminar session.

Why did I select the algorithm to use?

- I chose this Hashlib dependency for three reasons:
  - I had use it before in my Introduction to Cyber Security Module
  - It did not have a salt function
  - I wanted to see if I could implement the salting dependency to a more complex program

```python
import hashlib
import uuid
import time

# This function allows for conditions to be set to validate password
def password_check(Password):

# Passwprd must be five or more characters
    if len(Password) < 5:
        print('length should be at least 5')
        val = False
# Password must have at least one numeral
    if not any(char.isdigit() for char in Password):
        print('Password should have at least one numeral')
        val = False
# Password must have one uppercase letter
    if not any(char.isupper() for char in Password):
        print('Password should have at least one uppercase letter')
        val = False
# Password must have one lowercase letter
    if not any(char.islower() for char in Password):
        print('Password should have at least one lowercase letter')
        val = False
    if val:
        return val

```

```python
# start of authentication process
# This function displays a landing page welcoming users to notebook application, and gives guidance on next steps to be taken
def Welcome():
    print("Welcome to your Notebook app")
    print("Please choose one of the following to access the Notebook app")
    # User selects which option they prefer
    print("Select 1 to Register")
    print("Select 2 to Login")
    print()
    # As this is a simple 1/0 logic operator, the boolean returns the value True
    while True:
        print()
        user = input("Input your selection here: ")
        if user in ['1', '2']:
            break
    if user == '1':
        Register()
    else:
        Login()
```

```
    def Register():
        print("Welcome to the Notebook registration portal")

        # User inputs name and password if first time using system
        Username = input("Please input your name: ")
        Password = input("Please input your password:")

        # hashlib constructs password hashing, and uuid adds salting
        salt = uuid.uuid4().hex
        hashedpassword = hashlib.sha256(salt.encode() + Password.encode()).hexdigest() + ':' + salt


        # Sets conditons for password validation. If condtion are not met, then user is routed back to landing page
        if len(Password) < 5:
            print('length should be at least 5')
            val = False
        if not any(char.isdigit() for char in Password):
            print('Password should have at least one numeral')
            val = False
        if not any(char.isupper() for char in Password):
            print('Password should have at least one uppercase letter')
            val = False

        if not any(char.islower() for char in Password):
            print('Password should have at least one lowercase letter')
            val = False

    # This function opens the text file to add new users' username and hashed password
        file = open("D:\CODING\Parent\Jonnyash\SSD\demo.txt", "r")

    # This function compares users' username and password against the registered users'credentials, If already regsitered, output will state ' You are already registered'
        for f in file:
            if Username in f:
                print("You are already registered")

                Welcome()
                return
```

```
file.close()
    file = open("D:\CODING\Parent\Jonnyash\SSD\demo.txt", "a")
    file.write(Username + "," + hashedpassword)
    file.write(",")
    file.write(hashedpassword)
    file.write("\n")
    file.close()

# if newly registered and no exceptions raised, then output will state 'Your details have been entered'
    print("Your details have been entered")
    Welcome()
```

```
    def Login():

    # Welcome message asking for users' name and password
        print("Welcome to Queens Medical centre login portal")
        Username = input("Please input your name: ")
        Password = input("Please input your password:")

    # Starts the hash dependency for password decrypting
        Password, salt = hashedpassword.split(':')
        hashedpassword = hashlib.sha256(salt.encode() + Password.encode()).hexdigest()

    # This function opens the text file and compares against the registered users' credentials.
        with open ("D:\CODING\Parent\Jonnyash\SSD\demo.txt", "r") as file:
            for line in file:
                line = line.strip().split(",")
        if line[0] == Username and line[1] == hashedpassword:

    # If the users' username and password are already registered, then output will state 'You are now logged in'.
            print("You are now logged in")
            return True
        print("Sorry, your credentials are incorrect")

    # This time dependency has a three second duration before user can take the next step
        time.sleep(3)

    # This returns the user to the landing page to restart the process.
        return False


    Welcome()
    Register()
    Login()
```

```
Welcome to your Notebook app
Please choose one of the following to access the Notebook app
Select 1 to Register
Select 2 to Login


Input your selection here: 1
Welcome to the Notebook registration portal
Please input your name: James Bond
Please input your password:London007
Your details have been entered
Welcome to your Notebook app
Please choose one of the following to access the Notebook app
Select 1 to Register
Select 2 to Login
```

Figure 1. Registering snippet

```
1
2  James Bond,3848b923b74b981d1b18b626a558bcbc761a031daf60c0819fc4046fd033991e:6eb1d0e27ccd4acd926669644c5a4688,3848b923b74b981d1b18b626a558bcbc761a031daf60c0819fc4046fd033991e:6eb1d0e27ccd4acd926669644c5a4688
```

Figure 2. Hashed and Salted Password.

Would it meet GDPR regulations?

- GDPRUK states personal data must be secured (Ico.org.uk, 2019a)

- They specify encryption as an example, not as mandatory (Ico.org.uk, 2019a)

- GDPRUK does not specify much about passwords; however, personal data must be
- ‘Processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures’ (Ico.org.uk, 2019b)

References

- Ico.org.uk. (2019a) *Encryption* Available from: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/security/encryption/ [Accessed 31 Aug. 2022].
-
- Ico.org.uk. (2019b) *Passwords in online services*. Available from: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/security/passwords-in-online-services/ [Accessed 31 Aug. 2022].

- Jonny (2022) *1. Background*. GitHub. Available from: https://github.com/JonnyAsh/OOP_ASMIS/blob/main/ASMIS.py [Accessed 31 Aug. 2022].
- Tutorialspoint.com. (2020) *Python Modules of Cryptography*. Available from: https://www.tutorialspoint.com/cryptography_with_python/cryptography_with_python_modules_of_cryptography.htm [Accessed 31 Aug. 2022].